

BAB II

LANDASAN TEORI

2.1 Teori Dasar

Teori Dasar merupakan hal yang penting, untuk membuat penelitian menjadi jelas dan memiliki dasar yang kuat untuk dibahas dan sistematis. Menurut Sugiyono dalam (Iwan, 2019) Teori dasar perlu ditegakkan agar penelitian mempunyai dasar yang kokoh, dan bukan perbuatan coba-coba (*trial and error*), oleh karena itu teori yang akan digunakan pada penelitian ini adalah yang terkait seperti Android, Pelacakan, *GPS (Global Positioning System)*, *GIS (Geographic Information System)*, *UML (Unified Modeling Language)*, Perkembangan Perangkat Lunak, *Database*.

2.1.1 Android

Menurut (KOKAKOYUN, 2017) Android salah satu pemimpin pasar ponsel, memiliki satu miliar aplikasi di *Google Play Store* . Ini adalah sistem operasi seluler *open source* berbasis *Linux* yang dikembangkan oleh *Open Handset Alliance*, dan yang dikelola oleh *Google* untuk mengembangkan aplikasi untuk Android. Sistem operasi Android telah menjadi semakin populer, terutama di pasar *smartphone*. Ini adalah sumber terbuka dan disukai dalam penelitian ini karena fakta bahwa beberapa alat pengembangan gratis. Menurut (Chan, Andri Supriadi & Wahdi, Yera Wahda, 2018) Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Selain itu memungkinkan pengembang untuk menggunakan *platform* perangkat

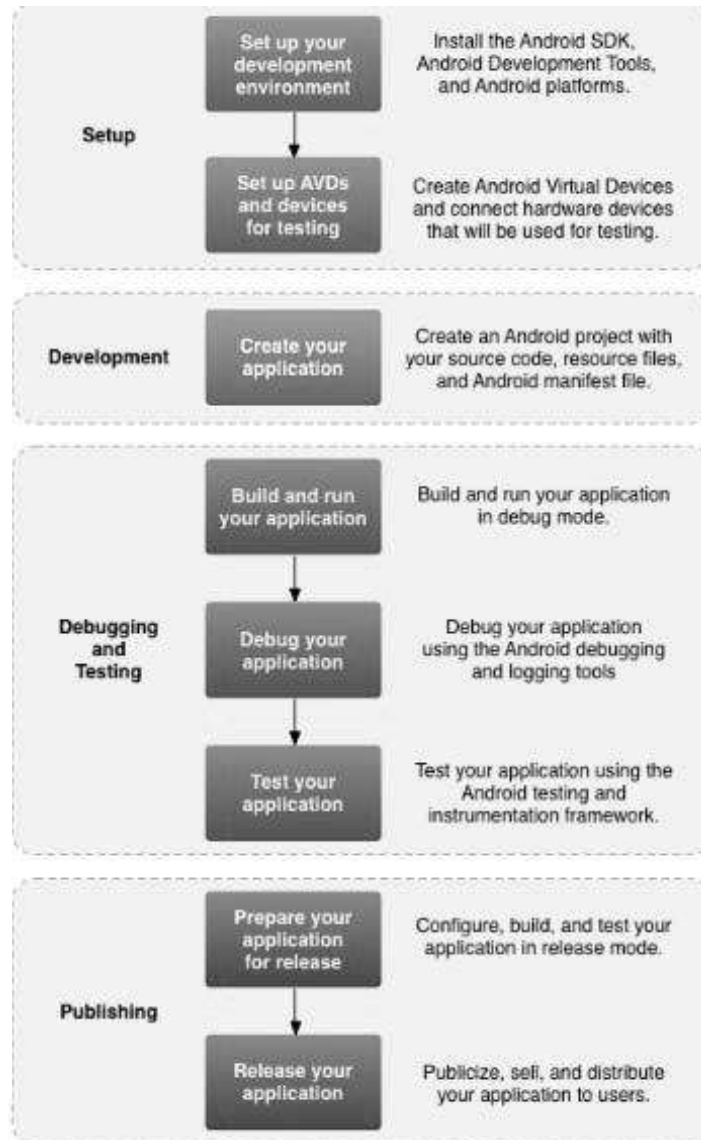
keras yang sangat cocok untuk sedikit upaya mewujudkan ide-ide mereka.

Sistem operasi ini pada awalnya di kembangkan oleh Android *Inc* kemudian dibeli secara resmi oleh *Google* pada tahun 2005.



Gambar 2.1 Logo Android
Sumber: Nadia Firly, 2018

Menurut (Lengkong, Sinsuw, & Lumenta, 2015) Android adalah *mobile operating system* yang mengadopsi sistem operasi *Linux* tetapi telah di modifikasi. Menurut Android dipuji sebagai *platform* seluler lengkap pertama, terbuka dan gratis. Menurut (KOCAKOYUN, 2017) Beberapa langkah yang harus diikuti saat mengembangkan aplikasi. Langkah-langkah dasar untuk mengembangkan aplikasi ditunjukkan pada Gambar 2.2.



Gambar 2.2 Bagian Pengembangan Aplikasi

Sumber: Kocakoyun, 2017

Dapat dilihat pada gambar 2.2 yaitu bagian pengembangan aplikasi yang di bagi menjadi 4 tahapan yaitu:

1. Pengaturan: Selama fase ini penelitian ini memasang dan mengatur lingkungan pengembangan kami. Penelitian ini juga membuat perangkat *virtual* Android dan menghubungkan perangkat perangkat keras, di mana penelitian ini dapat menginstal aplikasi.

2. Pengembangan: Selama fase ini penelitian ini menyiapkan dan mengembangkan proyek Android , yang berisi semua kode sumber dan *file* sumber daya untuk aplikasi pada penelitian ini.
3. *Debugging* dan *Testing*: Selama fase ini penelitian ini membangun proyek agar menjadi paket *debuggable.apk* yang dapat diinstal dan dijalankan pada emulator atau perangkat yang diberdayakan Android.
4. Menerbitkan: Selama fase ini penelitian ini mengkonfigurasi dan membangun aplikasi untuk rilis dan mendistribusikan aplikasi kepada pengguna.

Menurut (Rofiq & Uzzy, 2014) Fitur-fitur yang dimiliki Android adalah sebagai berikut:

1. *Framework* aplikasi, untuk melakukan daur ulang dan penggantian komponen.
2. Browser terintegrasi yang menggunakan *Engine Open Source Webkit* ,*Engine* ini juga digunakan pada browser safari *iPhone*.
3. Android menggunakan *Video Graphic Adapter* yang berkemampuan untuk mengolah grafik *2D* dan *3D*.
4. Android *support* berbagai teknologi konektivitas seperti *EDGE* (*Enhanced Data rates for GSM Evolution*), *UMTS* (*Universal Mobile Telecommunications System*), *CDMA* (*Code Division Multiple Access*), *EV-DO* (*Evolution-Data Optimized*), *GSM* (*Global System for Mobile Communications*), *Bluetooth* dan *Wi-Fi* (*Wireless Fidelity*).

5. Android dapat mengeksekusi *software* yang ditulis dalam bahasa Java menggunakan mesin virtual *Dalvik* yang dipersiapkan untuk penggunaan perangkat *mobile*.
 - a. Menyerupai dengan *App Store* pada *iOS*, Android memiliki *Google Play Store* yang digunakan untuk mengunduh aplikasi pada *smartphone* tanpa menggunakan komputer.
 - b. Android memiliki fitur *Multi touch* sehingga pengguna dapat menggunakan lebih dari satu jari saat berinteraksi dengan layer sentuh dalam waktu bersamaan. Android memiliki lingkungan dalam melakukan pengembangan, seperti *Debugging tools*, *Emulator* dan *Plugin* untuk *Eclipse IDE*.

Berikut merupakan versi sistem Android yang akan mendukung jalannya aplikasi ini:

1. Android 6.0 dan 6.0.1 "*Marshmallow*"

Marshmallow merupakan versi sistem operasi terbaru untuk Android *mobile*. *Marshmallow* akan fokus pada peningkatan pengalaman pengguna secara menyeluruh dan membawa fitur-fitur seperti model izin yang dirancang ulang agar aplikasi tidak lagi memberikan izin yang ditentukan secara otomatis dalam melakukan instalasi.

Gambar dibawah ini merupakan logo dari sistem operasi Android *Marshmallow*



Gambar 2.3 Logo *Marshmallow*
Sumber: Nadia Firly, 2018

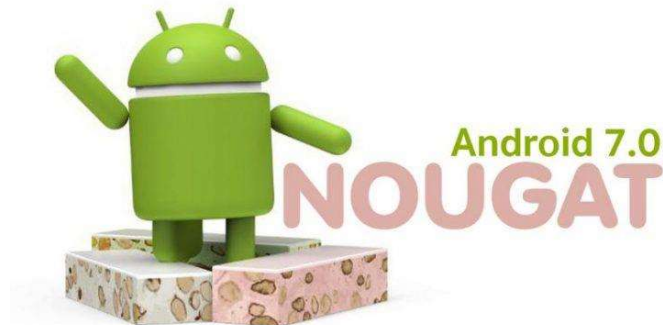
Android selalu merevisi sistem operasinya agar memenuhi kebutuhan pasar, di era sekarang ini teknologi semakin baru sehingga kebutuhan akan semakin meningkat. Hal tersebut terlihat dari banyaknya versi demi versi yang sudah dirilis oleh Android. Android telah dijadikan raja platform *smartphone* karena banyaknya fitur yang ditawarkan kepada pengguna.

2. Android 7.0 dan 7.1 "*Nougat*"

Nougat dirilis pada tanggal 22.08.2016. Android *Nougat* mengalami peningkatan dari versi Android sebelumnya. Fitur dukungan *multi-window* menjadi yang paling menarik dalam versi ini karena adanya pembaruan aplikasi sistem operasi yang lebih menarik dan halus sehingga aplikasi yang ada pada perangkat tersebut jadi lebih ringan.

- a. Memberi dukungan pada *smartphone* yaitu membuka dua aplikasi sekaligus.
- b. Terdapat fitur langsung untuk membalas pesan dari panel notifikasi.
- c. Desain yang lebih menarik khusus nya pada *panel notifikasi*.

Gambar dibawah ini merupakan logo dari sistem operasi Android *Nougat*



Gambar 2.4 Logo Android *Nougat*

Sumber: Nadia Firly, 2018

3. Android 8.0 "*Oreo*"

Android versi ini memiliki perbedaan dari versi sebelumnya, Android 8.0 lebih fokus pada kecepatan dan efisiensi dalam penggunaan harian, versi Android ini secara resmi dirilis pada pertengahan tahun 2017.

- a. Dapat menonton video saat menggunakan aplikasi lain lebih.
- b. Baterai tidak mudah boros.
- c. Emoji yang di *update* dan lebih banyak.

Gambar dibawah ini merupakan logo dari sistem operasi Android *Oreo*



Gambar 2.5 Logo Android *Oreo*

Sumber: Nadia Firly, 2018

Menurut (Nadia, 2019) Berikut adalah tabel versi-versi Android yang telah dirilis:

Tabel 2.1 Versi Android

Android 6.0	<i>MarshMallow</i>	5 Oktober 2015
Android 7.0	<i>Nougat</i>	9 Maret 2016
Android 7.1	<i>Nougat</i>	19 Oktober 2016
Android 8.0	<i>Oreo</i>	21 Maret 2017

Sumber: Nadia Firly, 2019

Menurut (Winata & Setiawan, 2013) Penjelasan arsitektur komponen utama dari sistem operasi Android adalah sebagai berikut:

1. *Linux Kernel* adalah tumpukan paling bawah dalam arsitektur Android. Android menggunakan *kernel linux* untuk membuat sistem Android. *Kernel* ini berfungsi untuk *abstraction* layer antara *hardware* dan *software*.
2. Android *Runtime* terdapat *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries* berisi inti *library* Java. *Dalvik Virtual Machine* memberikan fungsi kekuatan dan mengoptimalkan sistem operasi Android.
3. *Libraries* terdapat pada lapisan yang sama dengan Android *runtime* dan merupakan tempat yang menyimpan *set-set library*.
4. *Applications* dalam lapisan ini terdapat fungsi-fungsi pada *smartphone* seperti menelepon, mengirim pesan singkat, menjalankan web browser, melakukan akses pada kontak, melakukan permainan, dan lain-lain.

2.1.2 Pelacakan

Menurut (J. Andi, 2015) Pelacakan merupakan mengikuti jalan atau suatu kegiatan yang berguna mengikuti jejak suatu objek. Pengertian pelacakan dalam hal ini adalah kegiatan untuk melakukan pemantuan keberadaan posisi kendaraan pengangkut barang yang di dapatkan dari *smartphone user* (supir truk).

Pengertian pelacakan dalam hal ini adalah kegiatan untuk memantau keberadaan kendaraan berdasarkan posisi yang di dapatkan dari fitur *GPS (Global Positioning System)* pada *smartphone*.

Menurut (Nuchvi, Adi, & Mukhsin, 2015) Keuntungan dalam pelacakan terdiri dari:

1. Kendaraan minim resiko dari tindak kejahatan pencurian mobil karena dapat dilihat oleh pemilik.
2. Kendaraan dari pemilik mobil mengetahui lokasi dari kendaraan mereka.
3. Dapat mengirimkan koordinat kendaraan secara *realtime*.
4. Mempermudah karena adanya peta yang dapat menunjukkan kendaraan secara *real*.

Menurut Kompasiana (2018) Cara kerja dari pelacakan tergantung dari pengguna memilih sistem yang paling sesuai dengan kebutuhannya, setiap kendaraan akan mengirim data melalui jaringan seluler. Data dari koordinat akan dikirim ke *database* kemudian akan dikirim kembali ke aplikasi dalam tampilan *Google Maps*.

2.1.3 *Global Positioning System (GPS)*

Menurut (Susilo, Pranjoto, & Gunadhi, 2014) *Global Positioning System (GPS)* yang merupakan sebuah sistem yang dapat memastikan posisi dan sistem navigasi secara umum. Untuk menentukan koordinat objek yang di amati minimal memerlukan 4 buah satelit. *GPS* sebuah sistem navigasi yang dapat dipakai untuk memberikan informasi lokasi keberadaan seseorang berada dipermukaan bumi dengan bantuan satelit. Menurut (Faya & Fiftin, 2013) *GPS* pada awalnya digunakan pada kepentingan militer, tetapi pada tahun 1980an sudah bisa digunakan untuk kepentingan sipil, posisi *GPS* juga di tentukan berdasarkan titik-titik koordinat *latitude* dan *longitude*.

Menurut (Gunawan Desvandri, Darmo, & Bandi, 2016) Perkembangan *GPS* dan kegunaanya sejalan dengan kebutuhan manusia dalam macam macam aplikasi dan dalam keperluan penelitian. Menurut (Junus, 2012) Cara kerja sistem *GPS* menggunakan sejumlah satelit yang berada di orbit bumi dengan cara memancarkan *signal* ke bumi kemudian ditangkap oleh alat *receiver*.

Menurut (Suryanto, 2012) Langkah langkah cara kerja sistem *GPS* :

1. Memakai perhitungan *Triangulation* dari satelit.
2. Perhitungannya *GPS* dapat mengukur jarak menggunakan *travel time* sinyal radio.
3. Menghitung *travel time*, waktu dapat memberikan akurasi yang tinggi dalam *GPS*.
4. Mengukur jarak, yang perlu kita tahu adalah posisi satelit dan ketinggian pada orbitnya.

5. Pengkoreksian *delay* sinyal waktu perjalanan di atmosfer sampai di terima *receiver*.

Menurut (Alfeno & Devi, 2017) *GPS* juga dapat memberikan informasi lain seperti tujuan perjalanan, arah yang ingin dituju, jarak tujuan, matahari terbenam dan matahari terbit. Sinyal yang dikirimkan dari satelit ke *GPS* akan digunakan untuk menghitung waktu perjalanan. Dari informasi di atas *GPS* dapat di kategorikan menjadi:

1. Waktu: Sistem *GPS receiver* menerima informasi waktu dari jam atom yang memiliki keakuratan sangat tinggi.
2. Arah perjalanan: *GPS* bisa menunjukkan arah tujuan dan dapat menyimpan lokasi tempat yang sudah pernah dikunjungi di dalam *GPS receiver*.
3. Kecepatan: Ketika pengguna berpindah tempat, *GPS* dapat menunjukkan informasi mengenai kecepatan perpindahan tersebut.
4. *Tracking*: Untuk memonitoring pergerakan objek yang ingin di *tracking*.
5. Lokasi: *GPS* memberikan informasi lokasi seperti: *Latitude*, *Longitude*, *Altitude*.

2.1.4 Geographic Information System (GIS)

Geographic Information System (GIS) Menurut (Irwansyah, 2013) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis.

Menurut (Ependi, 2018) Di Indonesia pertama kali *GIS* dikenal pada tahun 1972 dengan nama data *banks for development*. *GIS* pertama kali diperkenalkan oleh *Assembly* yang berasal dari *International Geographical Union* di *Ottawa Canada* pada tahun 1967.

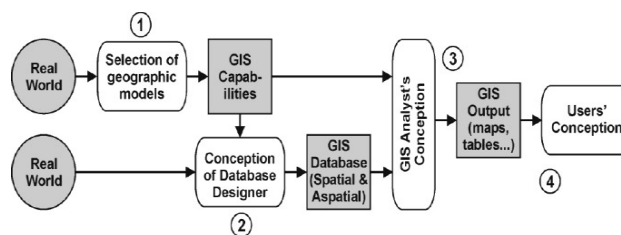
Sistem informasi geografis dapat disimpulkan sebagai gabungan kartografi, analisis statistik dan teknologi sistem basis data (*database*). Pendapat ahli menurut Aronoff dalam (Irwansyah, 2013) *GIS* merupakan jenis prosedur yang berbasis manual maupun komputer untuk menyimpan dan memanipulasi data bereferensi geografis.

Menurut Esri dalam (Asnawati & Putra, 2011) *GIS* merupakan sebuah sistem untuk mengatur, menganalisa dan menampilkan informasi geografis.

Berdasarkan pengertian *GIS* di atas dapat di rangkum sebagai berikut:

1. *GIS* merupakan informasi mengenai tempat di permukaan bumi.
2. Teknologi yang dipakai dalam *GIS* merupakan *GPS*, *remote sensing*.
3. *GIS* merupakan sistem komputer dan perangkat lunak.
4. *GIS* dapat digunakan untuk berbagai macam aplikasi.

GIS dalam melakukan komunikasi memiliki polanya sendiri.



Gambar 2.6 Model Alur Komunikasi *GIS*

Sumber: Ependi, 2017

Gambar 2.6 merupakan model alur komunikasi *GIS* yang terdiri dari 4 alur yaitu:

1. Memilih model geografis.
2. Konsep Perancang Basis Data.
3. Konsepsi analisis *GIS*.
4. Konsepsi Pengguna..

Menurut Budianto dalam (Yuwono, Aribowo, & Setyawan, 2015) Sistem Informai Geografis adalah gabungan dari 3 unsur pokok meliputi sistem, informasi dan geografis.

Beberapa komponen yang membangun sebuah *GIS* yaitu:

1. *Software*: Perangkat lunak terdiri dari instruksi dan kode yang disusun dengan cermat dalam berbagai bahasa komputer khusus.
2. *Hardware*: Ini merujuk pada bagian fisik komputer dan perangkat terkait.
3. *People*: Para pemangku kepentingan dilibatkan pada berbagai tahapan siklus kehidupan suatu sistem informasi, seperti pengguna akhir, spesialis, pemrogram, administrator basis data.
4. *Database*: basis pengetahuan yang berisi data.
5. *Network*: Media komunikasi dan dukungan jaringan adalah komponen vital dari suatu sistem informasi.

Sistem Informasi Geografis menyimpan informasi spasial tentang dunia nyata sebagai kumpulan lapisan tematik, di mana lapisan terkait bersama dengan geografi terkait. Konsep sederhana namun sangat kuat dan serbaguna ini telah terbukti sangat berharga untuk menyelesaikan banyak masalah di dunia nyata.

Kemampuan *GIS* untuk menangani dan memproses data yang direferensikan secara geografis membedakannya dari sistem informasi lainnya.

Tipe data adalah metode klasifikasi yang membedakan berbagai jenis data yang digunakan oleh sistem komputer. Manusia dapat dengan mudah mengenali berbagai jenis data dan menggunakan simbol khusus seperti \$ dan % untuk mewakili data. Demikian pula, sistem komputer menggunakan kode internal khusus untuk melacak berbagai jenis data yang mereka proses. Data yang direferensikan secara geografis menggambarkan lokasi dan karakteristik fitur spasial di permukaan bumi. *GIS* mendukung dua tipe data spasial dasar yaitu *Raster* dan *Vektor*.

Jenis data *raster* mewakili informasi spasial (geografis) dengan membaginya menjadi sel-sel yang diberi spasi dan dikuantisasi secara teratur. Sel adalah kotak kecil dan dikenal sebagai piksel (elemen gambar). Sel raster diatur sebagai matriks baris dan kolom.

Setiap piksel memiliki dua nilai terkait:

1. Lokasi piksel direpresentasikan sebagai nomor baris / kolom.
2. Nilai sel untuk mewakili atribut / properti yang menarik.

Berikut ini adalah contoh tipe data *vector*:

3. Poin mewakili titik diskrit di permukaan bumi.
4. Garis mewakili fitur linier seperti jalan. Setiap garis memiliki beberapa titik koordinat yang mempertahankan bentuknya.
5. Poligon mewakili area berikat seperti *waterbodies*.

2.1.5 *Unified Modeling Language (UML)*

Bahasa pemodelan terpadu adalah bahasa visual untuk mempresentasikan sistem melalui pemakaian diagram dan taks pendukung. Dengan *Unified Modeling Language (UML)*, model untuk perangkat lunak yang dapat di jalakan ke perangkat keras dapat dibuat. *UML* cocok digunakan untuk bahasa pemrograman *C++*, *C#*, dan Java karena menggunakan *class* dan *operation*.

Berikut tujuan atau fungsi dari *UML* :

1. Mempersiapkan model yang sudah siap untuk digunakan.
2. Membuat bahasa pemrograman yang dapat digunakan dan dimengerti oleh manusia ataupun mesin.
3. Dapat digunakan sebagai kerangka kerja terperinci.

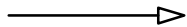
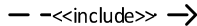
Selain tujuan dan fungsi, *UML* memiliki beberapa jenis diagram sebagai berikut:

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*

1. *Use Case Diagram*

Use Case Diagram merupakan salah satu dari diagram *Unified Modelling Language* yang mendeskripsikan hubungan interaksi lebih dari satu aktor dan sistem yang digunakan. Dibawah ini merupakan simbol dari *Use Case* yang di ambil dari buku Rekayasa Perangkat Lunak.

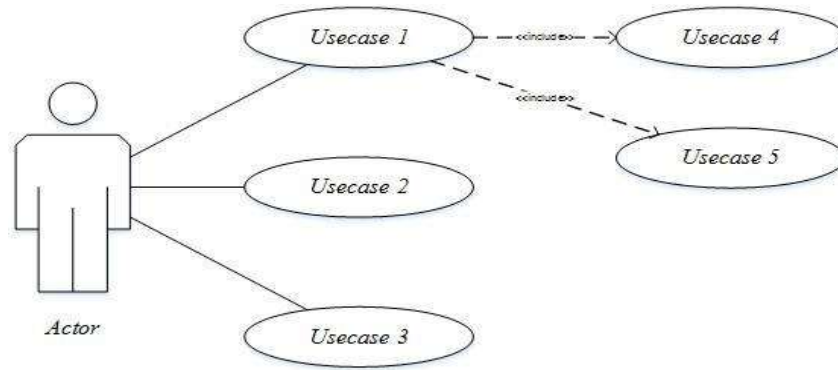
Tabel 2.2 Simbol *Use Case* Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan yang pengguna mainkan pada waktu berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan perubahan yang terjadi pada suatu elemen mandiri dan akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
3		<i>Generalization</i>	Hubungan antara objek anak berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk
4		<i>Include</i>	Memspesifikasikan <i>use case</i> di diagram sumber secara eksplisit
5		<i>Exclude</i>	Memspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber: Urva & Siregar, 2015

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah contoh perancangan simbol–simbol dalam *use case* diagram dapat digambarkan dalam contoh sebagai berikut:



Gambar 2.7 *Use Case Diagram*

Sumber: Urva & Siregar, 2015

2. Activity Diagram

Activity Diagram merupakan salah satu jenis *Unified Modelling Language* yang memodelkan semua proses yang terjadi dalam sistem. Terlihat pada tabel dibawah ini:

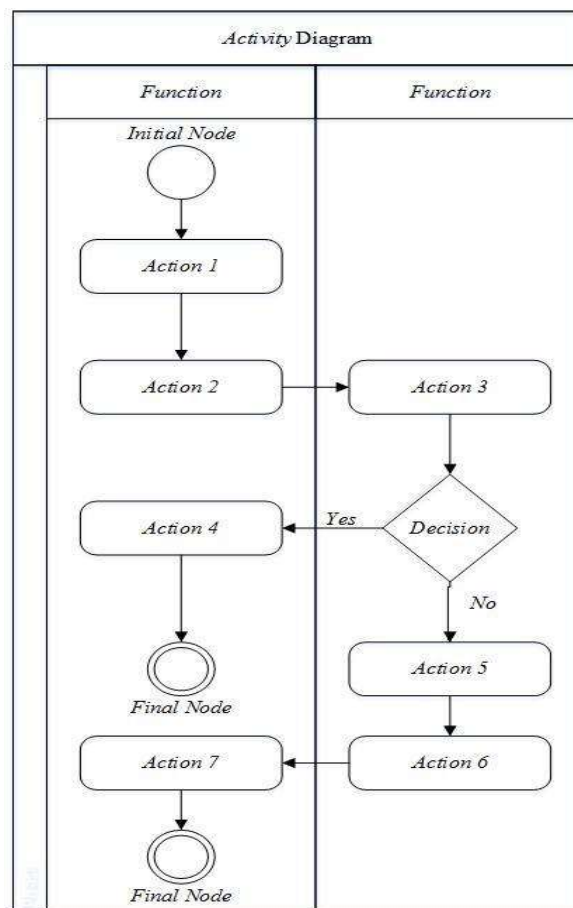
Tabel 2.3 Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
2		<i>Initial Node</i>	Bagaimana suatu objek dibentuk atau diawali
3		<i>Final Node</i>	Bagaimana suatu objek dibentuk dan diakhiri
4		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu

Sumber: Urva & Siregar, 2015

Menurut (Haqi & Setiawan, 2018) Diagram aktivitas merupakan salah satu jenis *Unified Modelling Language* yang menggambarkan pola semua proses yang berlaku dalam sistem. Diagram aktivitas juga menjelaskan proses paralel yang berjalan sepanjang eksekusi.

Simbol – simbol *Activity Diagram* jika digunakan dapat digambarkan dalam contoh sebagai berikut:

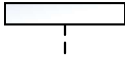


Gambar 2.8 Activity Diagram
Sumber: Urva & Siregar, 2015

3. Diagram *Sequence*.

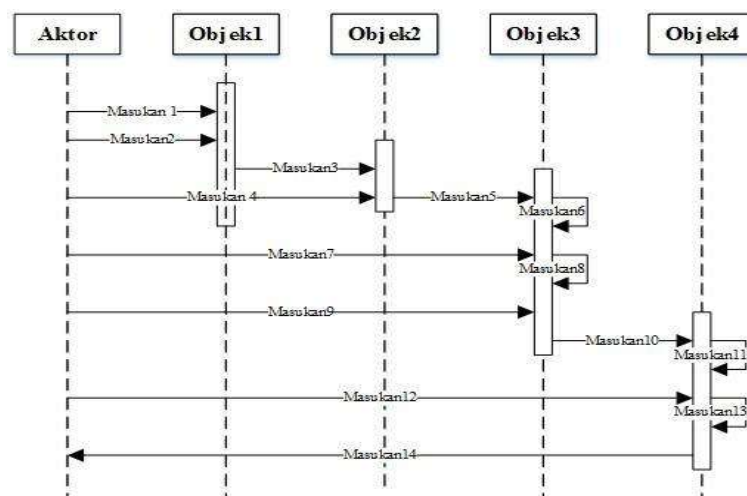
Diagram *Sequence* merupakan diagram pada *Unified Modelling Language* yang menjelaskan hubungan interaksi sesuai dengan rangkaian waktu.

Tabel 2.4 Simbol *Sequence* Diagram

No	Gambar	Nama	Keterangan
1		<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi antara satu sama lain.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek, dan memuat informasi tentang aktivitas yang terjadi.
3		<i>Activation</i>	Menyatakan waktu aktif.
4		<i>Self Message</i>	Menunjukkan bahwa relasi ini akan memanggil dirinya sendiri.

Sumber: Urva & Siregar, 2015

Simbol-simbol Diagram *Sequence* tersebut jika digunakan dalam diagram:



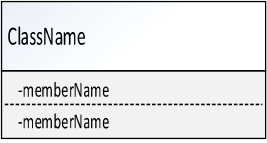
Gambar 2.9 Diagram *Sequence*

Sumber: Urva & Siregar, 2015

4. Class Diagram

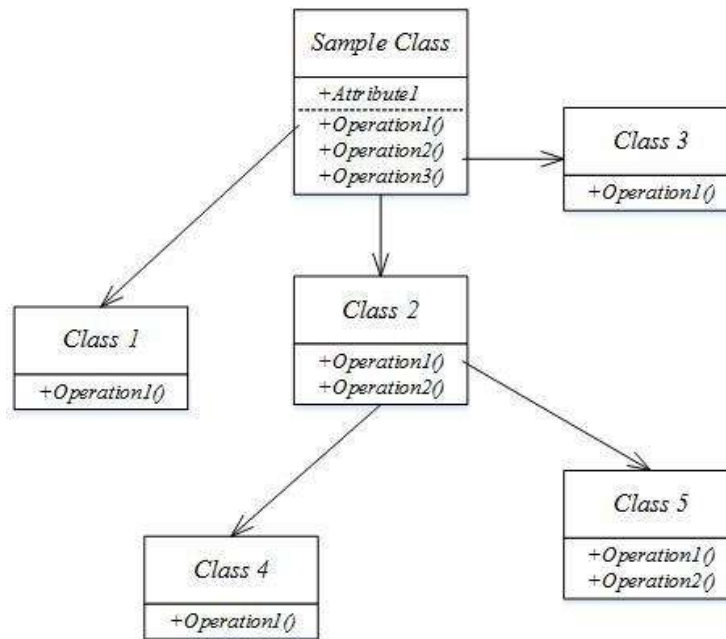
Class Diagram adalah diagram pada *Unified Modelling Language* yang menunjukkan kelas-kelas yang terdapat didalam sebuah sistem yang akan digunakan. Berikut merupakan simbol-simbol dari *Class Diagram*:

Tabel 2.5 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Class</i>	Menyatakan hubungan antara model kelas dalam sistem
2		<i>Association</i>	Merupakan penggambaran kelas yang mempunyai atribut kelas lain.
3		<i>Directed Association</i>	Asosiasi yang digunakan satu kelas digunakan oleh kelas lain juga.
4		<i>Dependency</i>	Kebergantungan atau relasi antara satu kelas dengan kelas lain.
5		<i>Aggregation</i>	Bentuk satu kelas dijadikan sebagai atribut kelas lain.
6		<i>Composition</i>	Bentuk lanjutan dari <i>Aggregation</i> , setelah kelas mejadi <i>whole</i> dibuat kelas yang menjadi bagian akan diciptakan.

Sumber: Urva & Siregar, 2015

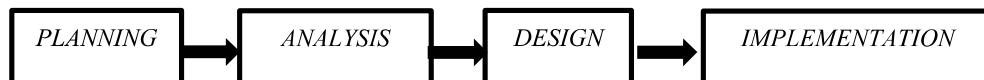
Berikut merupakan contoh perancangan diagram kelas yang dapat dilakukan :



Gambar 2.10 Class Diagram
Sumber: Urva & Siregar, 2015

2.1.6 Pengembang Perangkat Lunak

Pengembang perangkat lunak / *Software Development* adalah tipe proyek informasi teknologi yang berpusat pada pembuatan ataupun pengembangan perangkat lunak. *Software Development* berpusat pada sesuatu yang tidak nampak atau *software* sehingga proses pengerjaannya lebih sukar. Ada beberapa tahapan dalam *software development*, yaitu :



Gambar 2.11 Tahapan *Software Development*
Sumber: Bolung & Tampangela, 2017

Gambar 2.11 merupakan tahapan *Software Development*, berikut merupakan pengertian dari tahapan- tahapan *Software Development*:

1. *Planning*

Planning adalah tahap paling awal dalam memulai *software development* yang bertujuan untuk menghasilkan proses kerja yang jelas dan anggaran dana.

2. *Requirement dan Specification*

Tahap penentuan fitur-fitur yang dibutuhkan dalam perangkat lunak yang akan diciptakan.

3. *Architecture dan Design*

Tahap desain *interface*, *database* konseptual dan sistem keamanan dari perangkat lunak.

4. *Implementation dan Testing*

Setelah berpedoman pada tahap-tahap sebelumnya, maka dilakukanlah tahap implementasi. Setelah itu dilanjutkan oleh tahap *testing* yang merupakan uji coba kapasitas perangkat lunak.

5. *Deployment dan Maintenance*

Tahap terakhir adalah *Deployment dan Maintenance* agar perangkat lunak mulai dipakai pengguna. Ada dua aspek yang penting yaitu pelatihan penggunaan perangkat lunak dan pemantauan perangkat lunak.

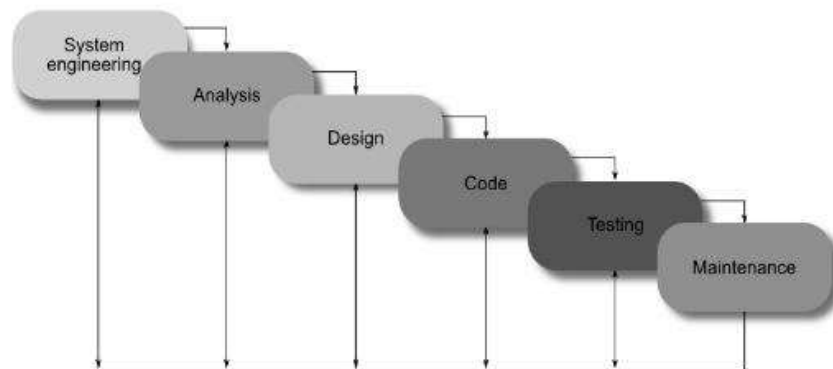
Menurut (Bolung & Tampangela, 2017) Komponen yang merupakan perkembangan dari perangkat lunak dapat di bagi dalam tiga bagian yaitu:

1. Metode: Menggunakan teknik pendekatan yang secara sistematis

berguna dalam mengembangkan perangkat lunak. Dalam hal ini metode melingkupi, uraian keperluan sistem dan perangkat lunak, perencanaan proyek, arsitektur program, penulisan *coding*, *testing*, prosedur algoritma dan pemeliharaan.

2. *Tools*: Alat yang mendukung dalam perkembangan perangkat lunak.
3. Prosedur: Dapat digunakan untuk mengartikan urutan pekerjaan dari metode yang dipakai dan alat bantu yang digunakan.

1. Model *Waterfall*



Gambar 2.12 Metode *Waterfall*
Sumber: Tampangela, 2017

Gambar 2.12 adalah rangkaian dari metode *Waterfall*. Model ini diperkenalkan oleh Winston Royce pada tahun 1970. Model *Waterfall* adalah metode pengembangan dalam perangkat lunak dengan cara pendekatan sekuensial dengan bagian sebagai berikut:

1. *System Engineering*: Perangkat lunak merupakan bagian dari sistem yang memiliki kapasitas cukup besar, pekerjaan berawal dari pembentukan kebutuhan kepada seluruh elemen pada sistem dan kemudian memilih

yang cocok untuk pengembangan. Hal ini cukup penting karena perangkat lunak berhubungan dengan hardware, orang dan basis data.

2. *Analysis*: Kumpulan dari perangkat lunak seperti domain informasi, fungsi yang dibutuhkan, untuk kerja dan antar muka. Kemudian hasilnya harus di dokumentasi dan di *review* ke pengguna.
3. *Design*: 4 bagian untuk program yang meliputi: struktur data, arsitektur perangkat lunak, prosedur *detail*, dan karakteristik antar muka. proses *design* dapat mengubah kebutuhan menjadi bentuk yang dimengerti oleh perangkat lunak sebelum penulisan program di mulai.
4. *Coding*: Menterjemah perancangan dalam bentuk yang dimengerti oleh mesin dengan menggunakan bahasa pemograman.
5. *Testing*: Jika sudah selesai dalam pembuatan *coding* maka *testing* harus dilakukan. *Testing* fokus kepada logika internal dari perangkat lunak, mencari adanya kemungkinan kesalahan dan memeriksa apakah sudah sesuai dengan yang diinginkan.
6. *Maintenance*: Bagian paling akhir dari *software development* dan dilakukan setelah perangkat lunak dipergunakan.

Tabel 2.6 Dampak Metodologi

Metodologi	Kelebihan	Kekurangan
<i>Waterfall</i>	1) Mudah dalam pengelolaan karena hampir seluruh persyaratan telah di identifikasikan dan di dokumentasikan. 2) Tahapan yang berurutan secara linier, identifikasi dan dokumentasi yang lengkap, menyebabkan proses mudah dipahami oleh seluruh yang terlibat.	1) Tahapan yang berurutan secara linier tidak memungkinkan untuk kembali pada tahapan selanjutnya. 2) Tidak fleksibel terhadap perubahan kebutuhan yang terjadi dalam tahap pengembangan sistem. 3) Hampir tidak ada toleransi kesalahan, terutama pada tahapan rencana dan desain.

Sumber: Tampangela, 2017

2.1.7 Database

Database atau *Basis data* merupakan sebuah tempat penyimpanan data dalam komputer yang bersifat sistematis. Proses perancangan sebuah aplikasi dibutuhkan sebuah tempat penyimpanan yang dapat diakses dari sisi server yang memakai bahasa komputer seperti *PHP (Hypertext Preprocessor)* dan *ASP (Active Server Pages)* dan untuk melakukan pengubahan data baik itu *read, create, update* atau *delete* data membutuhkan proses pengiriman *request* ke server (Ilhami.M, 2017). Ini akan menjadi tantangan untuk para *developer* karena perlunya ketersediaan *application programming interface* yang dapat memenuhi kriteria dengan baik oleh aplikasi yang akan dibangun. Berbagai macam *database* antara lain *MySQL(My Structured Query Language)*, *Oracle*, *SQL (Structured Query Language)*, *Server*, *Firebase* dan lain sebagainya.

Menurut (Hendini, 2016) *database* merupakan kumpulan dari tabel-tabel yang saling berelasi dan relasi tersebut dapat ditunjukkan dengan kunci dari setiap tabel yang ada. *Database* juga merupakan kumpulan data yang menggambarkan aktivitas dan pelakunya dalam suatu organisasi. Sistem *database* merupakan sistem komputer yang berguna untuk menyimpan dan mengelola data.

Basis data adalah sistem yang tujuan utamanya adalah menjaga data yang sudah digunakan dan membuat informasi tersedia saat diperlukan. *Database* merupakan tempat untuk menyimpan data untuk dapat diakses dengan mudah. Kebutuhan *database* yaitu seperti memasukkan, menyimpan, dan mengambil data dan menerbitkan laporan yang bersumber pada data yang telah disimpan.

Bentuk *database* yang diperlukan dalam sistem yaitu (*DBMS*) *Database Management System*. *DBMS* adalah sistem aplikasi yang berfungsi untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut *Database Management System* jika mencakupi syarat seperti di bawah ini:

1. Tersedia fasilitas untuk mengakses akses data.
2. Kemampuan menangani integritas data.
3. Kemampuan menangani akses data yang dilakukan.
4. Kemampuan menangani *backup* data.

Menurut (Budi, John, & Soetarmono Anggya Nanda, 2015) *MySql* adalah *database server* yang cukup terkenal, *MySQL* berada di bawah lisensi *General Public License* dan bersifat *free*, maka dalam penelitian ini *mySQL* dapat digunakan dengan bebas tanpa perlu takut dengan lisensi yang sudah ada, jenis

dari *mySQL* merupakan jenis dari *Relational Database Management System*. Ini yang menyebabkan ada istilah *table*, baris, kolom yang dipakai pada *mySQL*.

Menurut (Yuliansyah,H 2014) *MySQL* adalah sebuah *database* manajemen system (*DBMS*) populer yang memiliki fungsi sebagai relational *database* manajemen system (*RDBMS*). Selain itu *MySQL software* merupakan suatu aplikasi yang sifatnya open source serta server basis data *MySQL* memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded system*. Dikarenakan faktor *open source* dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data.

2.2 Pelacakan Kendaraan

Menurut (J, Andi, 2015) Pelacakan / *Tracking* adalah mengikuti jalan atau suatu kegiatan yang berguna mengikuti jejak suatu objek. Pengertian *tracking* adalah kegiatan untuk melakukan pemantauan keberadaan posisi kendaraan pengangkut barang yang di dapatkan dari fitur *GPS (Global Positioning System)* pada *smartphone*.

Menurut (Kompasiana, 2018) ketika dilakukan pelacakan pada mobil perusahaan maka kita mendapatkan informasi mengenai lokasi kendaraan, kecepatan, arah perjalanan dan tentunya melacak karyawan.

Berikut adalah keuntungan ketika menggunakan aplikasi pelacakan pada kendaraan:

1. Membantu mencegah pencurian jika pencurian terjadi.
2. Dapat memberikan informasi *realtime* mengenai posisi kendaraan.

3. Dapat mendisiplinkan supir pengguna dapat memantau kinerja supir dengan contoh misalnya supir sering berhenti.

2.2.1 Pelacakan Kendaraan Pengangkut Barang

Penelitian ini dilakukan pada perusahaan PT. Pulau Bulan Indo Perkasa, latar belakang perusahaan ini bergerak di bidang kontraktor bagian jalan yang sudah berdiri lebih dari 10 tahun. Kantor dari perusahaan ini beralamat di Tanjung Pantun Blok R No.3, Sungai Jodoh, Kecamatan. Batu Ampar, Kota Batam. Untuk *workshop* nya berlokasi di Telaga Punggur. Perusahaan ini menjual dan melakukan pengiriman material yang di minta oleh penerima di lokasi tujuan. Material seperti aspal, batu *split*, batu *bouksit* dan *base course*.

Perusahaan memiliki 20 armada truk yang beroperasi dan 1 orang pengawas gudang. Keberadaan armada truk sering menjadi keingintahuan pengawas gudang maupun pemiliknya. Proses pengiriman material ke lokasi tujuan sering sekali terdapat permasalahan yang di timbulkan seperti keterlambatan kedatangan, jumlah barang pada lokasi tujuan berkurang. Hal demikian sering terjadi karena lemahnya pengawasan terhadap truk tersebut dan akan menimbulkan biaya operasional yang lebih bagi perusahaan. Sehingga dibutuhkan sistem yang dapat membantu tugas pengawas gudang dalam mengkontrol armada truknya.

Tujuan dalam penelitian ini yang akan dicapai yaitu adalah pelacakan truk dengan menggunakan *GPS (Global Positioning System)* pada perangkat telepon yang berfungsi untuk memberitahu lokasi truk kepada pengawas gudang. Untuk menghadapi permasalahan tersebut maka dalam penelitian ini dirancang suatu

aplikasi yang dapat berfungsi melacak posisi truk dan dapat dilihat oleh pengawas gudang melalui halaman *mobile*.



Gambar 2.13 Material Batu Granit
Sumber: Arsip PT. Pulau Bulan Indo Perkasa

Gambar 2.13 merupakan material stok batu granit yang dimiliki oleh PT. Pulau Bulan Indo Perkasa, material ini sering dibeli oleh perusahaan lain karena digunakan untuk campuran pembuatan *readymix* dan material *duststone*.



Gambar 2.14 Material Abu Dust
Sumber: Arsip PT. Pulau Bulan Indo Perkasa

Gambar 2.14 merupakan stok material *abu dust* yang dimiliki oleh PT. Pulau Bulan Indo Perkasa, material ini sering dibeli oleh perusahaan lain karena digunakan untuk campuran pembuatan material aspal.



Gambar 2.15 Material Pasir
Sumber: Arsip PT. Pulau Bulan Indo Perkasa

Gambar 2.15 merupakan stok material pasir yang dimiliki oleh PT. Pulau Bulan Indo Perkasa, material pasir ini sering dibeli oleh perusahaan lain karena digunakan untuk campuran pembuatan material *readymix*.



Gambar 2.16 Truk Pengangkut Barang
Sumber: Arsip PT. Pulau Bulan Indo Perkasa

Gambar 2.16 merupakan armada truk yang digunakan untuk mengangkut material pasir, batu granit, *abu dust*, dan aspal yang ada di PT. Pulau Bulan Indo Perkasa. Perusahaan ini memiliki total 20 truk pengangkut barang.



Gambar 2.17 Proses Loading Material Aspal
Sumber: Arsip PT. Pulau Bulan Indo Perkasa



Gambar 2.18 Proses Loading Material Aspal
Sumber: Arsip PT. Pulau Bulan Indo Perkasa

Gambar 2.17 dan 2.18 merupakan *Asphalt Mixing Plant* yang ada di PT. Pulau Bulan Indo Perkasa, alat ini merupakan proses dalam membuat aspal, material ini sering dibeli oleh perusahaan lain dikarenakan untuk membuat dan memperbaiki jalan yang ada di Kota Batam.

2.3 Software Pendukung

1. Google Maps

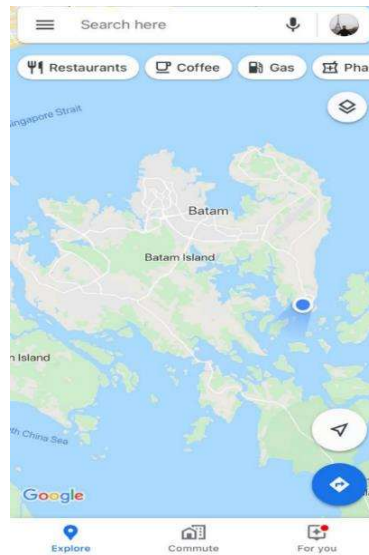
Menurut (Eka, Randy, & Yhan, 2012) *Google Maps* merupakan *platform* yang memberikan jasa peta gratis *online* yang tersedia oleh *Google* dan dapat ditemukan pada <http://maps.google.com/>. Menurut (Rismayani, 2016) *Google Maps* adalah peta dunia yang dapat digunakan untuk melihat suatu daerah dan layanan yang diberikan oleh *Google* secara gratis.

Menyediakan layanan berbasis peta yang sangat responsif dan ramah pengguna. Dengan menggunakan *Google Maps* ini, pengguna dapat dengan mudah menemukan lokasi tertentu dan mendapatkan tempat dengan mudah.



Gambar 2.19 Logo *Google Maps*
Sumber: Aplikasi *Google Maps*

Gambar dibawah ini merupakan tampilan *interface* dari *Google Maps*



Gambar 2.20 Tampilan *Interface Google Maps*
Sumber: Aplikasi *Google Maps*

Gambar 2.19 dan 2.20 merupakan logo dari *Google Maps* dan tampilan *interface* pada *smartphone*. Di tingkat pemrograman, *Google Maps* dapat dikembangkan dengan basis data, semua data yang terkait dengan lokasi disimpan dalam tabel dan dapat ditampilkan dengan mudah.

2. *Google Maps API (Application Programming Interface)*

Menurut (Garude Mihir & Nirmal, 2014) *Google Maps API (Application Programming Interface)* adalah aplikasi dan teknologi layanan pemetaan berbasis web yang disediakan oleh *platform Google*, yang mendukung banyak layanan berbasis peta.



Gambar 2.21 Logo *Google Maps API*
Sumber: Aplikasi *Google Maps API*

Gambar 2.21 merupakan logo dari *Google Maps API* dan tampilan *interface* yang dimiliki sama seperti tampilan pada *Google Maps*. *Google Maps JavaScript API* memungkinkan untuk menyematkan *Google maps* di halaman web sendiri. Dengan menggunakan *Google maps API*, dimungkinkan untuk menyematkan situs *Google maps* ke situs web eksternal, di mana data spesifik situs dapat dilapis. *Google Maps API* dapat memudahkan pengembang sistem dalam merancang dan mengimplementasikan sebuah media informasi.

3. Android Studio

Menurut (J. Andi, 2015) Android Studio merupakan *IDE (Integrated Development Environment)* resmi yang di rilis *Google* pada 16 Mei 2013. Sebelum adanya Android studio *software* pengembangan Android sebelumnya yaitu *Eclips*.

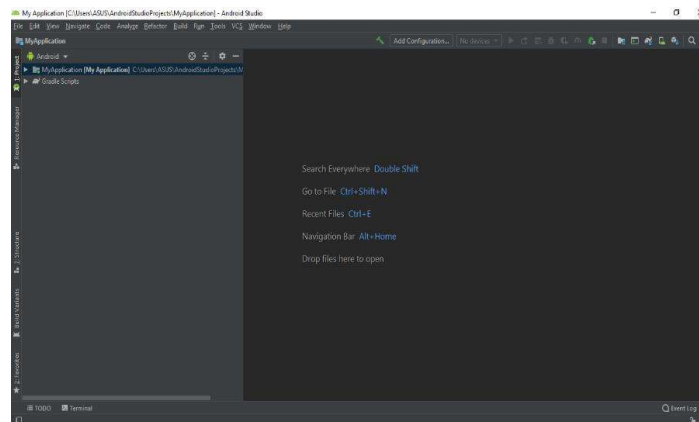


Gambar 2.22 Android Studio
Sumber: Aplikasi Android Studio

Gambar 2.22 merupakan logo dari Android Studio. Android *Studio* memiliki beberapa fitur yang berguna dalam membuat aplikasi Android seperti :

1. Proyek berbasis pada *Build Gradle*
2. Penanganan *bug* yang tepat.

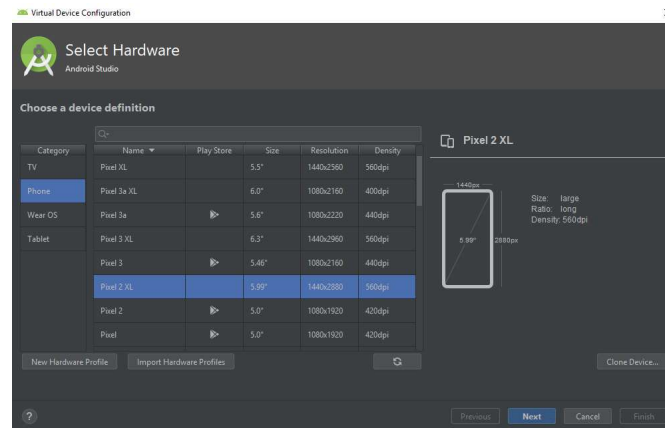
3. *Tools “Lint”* yang berfungsi untuk memonitor kecepatan, kegunaan, dan kompabilitas aplikasi dengan cepat.
4. Mendukung penandatanganan aplikasi untuk keamanan.
5. Mempunyai *GUI (Graphical User Interface)* yang membuat Android studio menjadi diminati dan membuat aplikasi menjadi lebih sederhana.
6. Didukung semua *Google Cloud Platform* untuk aplikasi yang dikembangkan.



Gambar 2.23 Tampilan *Interface* Android Studio
Sumber: Aplikasi Android Studio

Menurut (Kusniyati & Sitanggang, 2016) *Android SDK (Software Development Kit)* merupakan *tools* yang digunakan untuk mengembangkan aplikasi Android dengan menggunakan bahasa pemrograman Java.

Android Virtual Device (AVD) adalah emulator pada Android studio untuk menjalankan program aplikasi Android yang kita buat. *Android Virtual Device* memungkinkan Anda membuat, memperbarui, menghapus, perbaikan, dan menjalankan sebuah *AVD*.



Gambar 2.24 Tampilan *Interface Android Virtual Device*
Sumber: Aplikasi Android Studio

Sebelum menggunakan *AVD* (*Android Virtual Device*) harus ditentukan dahulu karakteristiknya contohnya dalam menentukan versi Android, ukuran layar, jenis layar dan besarnya memori.

Menurut (Maiyana, Efmi, 2018) Android Studio merupakan *IDE (Integrated Development Environment)* dalam pengembangan aplikasi Android. Aplikasi ini dipublikasikan oleh *Google* pada 16 Mei 2013.

Menurut (Maiyana, Efmi, 2018) Program komputer yang memiliki fasilitas yang diperlukan dalam *software development* dengan fungsi terintegrasi yang dibutuhkan untuk membangun sebuah perangkat lunak seperti *debugger*, *compiler*, *code editor*.

4. Java

Menurut (Kusniyati & Pangondian Sitanggang, 2016) Java adalah bahasa pemrograman yang dikembangkan oleh *SUN Micro System*. Untuk membuat suatu aplikasi Java, dibutuhkan *Compiler* dan *Interpreter* yang berbentuk *Java Development Kit (JDK)*.



Gambar 2.25 Logo Java
Sumber: Wahana Komputer, 2010

Instalasi *JDK* harus dilakukan terdahulu sebelum instalasi Android *Software Development Kit (SDK)*. Menurut (J. Andi, 2015) *Java Development Kit* digunakan untuk mengembangkan perangkat lunak yang berbasis Java.

5. *Graphical User Interface (GUI)*

Menurut (O. Andi, 2010) *GUI (Graphical User Interface)* adalah sebuah interaksi antara manusia dan komputer dalam melakukan suatu operasi komputer akan memberikan tampilan grafis yang lebih mudah dipahami oleh manusia. *GUI* adalah sebuah metode interaksi manusia dan komputer, komputer menyediakan tampilan grafis yang mudah dipahami oleh manusia dalam menggunakan suatu aplikasi. Java terdapat 2 teknologi dalam membuat tampilan berbasis grafis yaitu *AWT (Abstract Window Toolkit)* dan *Swing*.

Teknologi *AWT* mempunyai tampilan grafis yang bergantung secara penuh kepada sistem operasi, sedangkan *Swing* mempunyai tampilan grafis yang bersifat tidak tergantung pada sistem operasi. Terdapat teknologi *SWT (Widget Standar Toolkit)* yang dikembangkan oleh *IBM* dalam pemograman grafis.

Perbedaan penulisan dalam *AWT* dan *Swing* terletak pada awalan nama komponen itu, pada dasarnya teknologi *AWT* langsung memakai nama komponen,

sedangkan *Swing* diawali dengan huruf J. Misalnya komponen *AWT* adalah *TextField*, maka komponen *swing* adalah *JTextField*.

Tabel 2.7 Tampilan Berbasis Grafis

AWT	Swing	Keterangan
<i>Window</i>	<i>JWindow</i>	Top level <i>container</i> tanpa <i>border</i> , <i>title</i> bar, dan fungsi <i>management window</i> lainnya.
<i>Frame</i>	<i>JFrame</i>	Top level <i>container</i> yang dilengkapi <i>border</i> , <i>title</i> bar, dan lainnya. Banyak digunakan dalam pembuatan aplikasi <i>windows</i> .
<i>Dialog</i>	<i>JDialog</i>	Top level <i>container</i> yang dilengkapi <i>frame</i> akan tetapi digunakan sebagai <i>windows</i> sementara untuk mengambil input dari <i>user</i> .
Panel	<i>JPanel</i>	<i>Container</i> yang sering dipakai untuk memuat komponen lainnya. Panel lebih sering digunakan untuk mengelompokkan dan mengatur tata letak komponen

Sumber: Wahana Komputer, 2010

1. *Container*

Container adalah komponen yang digunakan untuk memuat komponen atau bahkan kontainer lain. *Container* juga dapat dikatakan sebagai *root* (akar) dari pohon hierarki komponen yang membentuk suatu aplikasi. *Container* dibagi menjadi 2 yaitu *container* level pertama (*Top Level Container*) dan *container* level kedua (*Second Level Container*). *Container* level pertama digunakan untuk membuat jendela (*Form*) yang tidak tergantung pada *container* lain atau tidak dimuat kontainer lain. Sedangkan pada *container* level kedua terdiri dari jendela yang dimuat pada *container* level pertama, *container* ini harus dimuat pada *container* level pertama dan dapat di tambahkan *container* level kedua dan komponen *GUI*.

2. Komponen

Komponen *GUI* dalam java terbagi menjadi 2 kelas besar, yaitu kelas *Component* dan kelas *Menu Component*. Kelas *Component* terdiri dari komponen-komponen kontrol *User Interface (UI)* seperti *Button*, *TextField*, *Checkbox*, *Choice*. Sedangkan kelas *Menu Component* terdiri dari menu, menu item yang digunakan untuk membuat menu *popup*.

6. Microsoft Visio

Microsoft Visio adalah program aplikasi komputer yang digunakan untuk membuat diagram, diagram alir, *brainstorm*, dan skema jaringan yang dibuat oleh *Microsoft Corporation*. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

2.4 Penelitian Terdahulu

Beberapa penelitian terdahulu yang menjadi referensi penelitian diantaranya yaitu :

1. Penelitian sebelumnya yang dilakukan oleh Y. Dul Muchlisin dan J. Eko Istiyanto, dalam penelitiannya yang berjudul **“Implementasi Sistem Pelacakan Kendaraan Bermotor Menggunakan GPS dan GPRS dengan Integrasi Google Map”**, VOL 5 NO.2, Juli, 2011. Penelitian ini memanfaatkan *GPS receiver* berupa *latitude* dan *longitude* kemudian diparsing ke website melalui koneksi *GPRS* dan data posisi yang dikirim dari *GPS receiver* dalam format *NMEA-0183*. Sehingga hasil yang di dapat pada penelitian ini adalah dapat melihat *history* posisi keberadaan

kendaraan selama perjalanan. Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah penelitian ini menggunakan metode *Waterfall* dan variabel yang digunakan.

2. Penelitian sebelumnya yang dilakukan oleh Rifai Ahmad, dalam penelitiannya yang berjudul **“Sistem Informasi Pemantauan Posisi Kendaraan Dinas Unsri Menggunakan Teknologi GPS”**, VOL 5, NO2, Oktober 2013, Rifai Ahmad menggunakan teknologi *GPS* dalam penelitiannya yang kemudian jika terjadi perpindahan posisi dari kordinat ke kordinat lainnya maka akan disimpan langsung ke *GPS tracking server*. Hasil yang diperoleh adalah ketika melakukan *refresh* data maka kita akan mendapatkan efek bergerak pada peta *Google Maps* Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah penelitian ini menggunakan metode *Waterfall* dan variabel yang digunakan.
3. Penelitian sebelumnya yang dilakukan oleh Sulisty Warjono dkk, dalam penelitiannya yang berjudul **“SISTEM PENGAMAN DAN PELACAK KENDARAAN BERMOTOR MENGGUNAKAN GPS DAN SMS”**, VOL.3 No.1 April 2014. Penelitian ini memanfaatkan teknologi *GPS* dan perangkat seluler untuk pelacakan dan pengamanannya, modul yang digunakan pada penelitian ini adalah *SIM548C* dan mikrokontroler *AT* serta relai sebagai pemutus kontak. Sehingga hasil yang di dapat pada penelitian ini adalah posisi kendaraan dapat diketahui dengan pasti dan kendaraan juga dapat dimatikan mesinnya dengan mengaktifkan relai lewat *SMS*. Perbedaan penelitian ini dengan penelitian yang akan

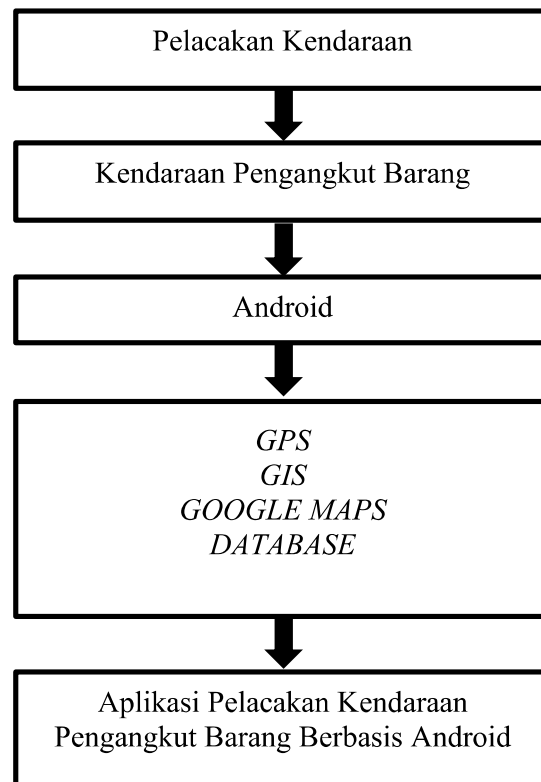
dilakukan adalah penelitian ini menggunakan metode *Waterfall* dan variabel yang digunakan.

4. Penelitian sebelumnya yang dilakukan oleh Indrajaya, Affandi, Pratomo yang berjudul **“Perancangan *Automatic Vehicle Location Tracking System* Berbasis Web *Mapping* Menggunakan *GPS* dan *GIS* Pada Kendaraan Angkutan Umum”**, NO 1 Tahun 2015 dalam penelitiannya, Indrajaya, Affandi, Pratomo menganalisis tracking sistem berbasis web menggunakan *GPS* pada kendaraan angkutan umum menggunakan metode algoritma *dijkstra* agar calon penumpang mendapatkan kendaraan yang terdekat dengan posisi mereka. Hasil yang diperoleh adalah ketika data dari proses pengujian sistem berhasil pembacaan *GPS* bisa terjadi *error* dalam hal pengiriman data terhadap performa sistem secara keseluruhan Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah penelitian ini menggunakan metode *Waterfall* dan variabel yang digunakan.
5. Penelitian sebelumnya yang dilakukan oleh Rian Affrilianto, Dedi Triyanto, Suhardi, dalam penelitiannya yang berjudul **“RANCANG BANGUN SISTEM PELACAK KENDARAAN BERMOTOR MENGGUNAKAN *GPS* DENGAN ANTARMUKA *WEBSITE*”**, VOL 05 No.3 2017. Penelitian ini memanfaatkan *GPS* dengan antar muka *website* yang dapat berfungsi mematikan mesin kendaraan motor dari jarak jauh dan memonitoring keberadaan posisi kendaraan motor dari halaman *website* dengan menggunakan *Arduino mega 2560* dan *GPS*

VK2828U7G5LF. Sehingga hasil yang di dapat pada penelitian ini adalah apabila kendaraan motor hilang maka pemilik kendaraan motor tersebut dapat mengetahui lokasi kendaraan dan dapat mematikan kendaraan dari jarak jauh. Perbedaan penelitian ini dengan penelitian yang akan dilakukan adalah penelitian ini menggunakan metode *Waterfall* dan variabel yang digunakan.

2.5 Kerangka Pemikiran

Pelacakan kendaraan pengangkut barang berbasis Android dengan permasalahan sering terjadi keterlambatan pengiriman material, kurang transparansi informasi sehingga dari permasalahan tersebut dapat membuat aplikasi pelacakan kendaraan pengangkut barang dengan bantuan teori dasar seperti Android, *GPS (Global Positioning System)*, *GIS (Geographic Information System)*, *UML (Unified Modeling Language)*, *Google Maps*, *Database* dan akan menghasilkan aplikasi sistem pelacakan kendaraan berbasis Android yang akan langsung di implementasikan ke *smartphone* supir truk. Terlihat pada gambar 2.26



Gambar 2.26 Kerangka Pemikiran
Sumber: Data Penelitian (2019)